UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/672,368	09/28/2000	Francis X. McKeen	042390.P9575	7652

7590 04/07/2005

Blakely Sokoloff Taylor & Zafman LLP
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025

EXAMINER	
HO, THOMAS M	

ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 04/07/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/672,368	MCKEEN ET AL.	
	Examiner	Art Unit	
	Thomas M Ho	2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on *07 September 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) *16-30* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☒ Claim(s) *16-30* are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-30 are pending.

2. The amendment of 9/7/04 has been received and entered.

Election/Restrictions

3. Newly submitted claims 16-30 are directed to an invention that is independent or distinct from the invention originally claimed for the following reasons:

Group I. Original claims 1-15 are directed towards methods comprising the structure and setup of page tables and memory mapping classified in 711/208 and 711/209, and their manipulation. (Memory Management Techniques: Segment or Page descriptor and logical address spaces, pages, segments, blocks). An Example of this is a memory mapper, or paging technique used for process control blocks (PCBs) such as demand paging and segmentation.

Group II. Claims 16-21, 29-30 are directed towards a method of event handling and recognition, classified in 719/318. (Event Handling and Event Notification). An example of this is a classic event handler such as that used in windows messaging systems.

Group III. Claims 22-30 are directed towards an apparatus concerning the a hardware implementation multiple modes of execution within a CPU classified in 712/43 (Processing Architecture: Mode Switching). An example of this is a secure processor or processor design.

Inventions I and II, II and III, I and III are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are shown to be separately usable.

In the instant case, invention I has separate utility such as a memory mapper.

Invention II has separate utility as an event handler. Invention III has separate utility as a particular processor design.

See MPEP § 806.05(d).

MPEP 803 states:

For purposes of the initial requirement, a serious burden on the examiner may be prima facie shown if the examiner shows by appropriate explanation either separate classification, separate status in the art, or a different field of search as defined in MPEP Section 808.02.

Because these inventions are distinct for the reasons given above and have acquired a separate status in the art as shown by their different classification and would require a different field of search restriction for examination purposes as indicated is proper.

Since applicant has received an action on the merits for the originally presented invention, this invention has been constructively elected by original presentation for prosecution on the merits. Accordingly, claims 16-30 are withdrawn from consideration as being directed to a non-elected invention. See 37 CFR 1.142(b) and MPEP § 821.03.

Response to Arguments

4. Coulouris discloses computer operating systems at a difference level of abstraction than that disclosed and claimed in the instant application, which makes meaningful comparisons difficult, but Applicant's offer several observations in reference to the Examiner's comments. First, as to the likening of an "isolated execution mode" to a "processor running a secure process," the cited portion of Coulouris does not discuss a processor running a secure process, nor is it readily apparent from the surrounding material what such a processor might be, or how it is like the claimed isolated execution mode.

The Examiner contends that while no processor is explicitly disclosed, it is evident there is a processor running the operating system and the isolated execution environment. As all operating systems run on digital systems, a processor would be virtually inherent to the execution of such a disclosure. However, the Examiner considers this point moot. No limitation of a "processor running a secure process" is found in claim 1.

As for the Applicant's objection to the Examiner's likening of an isolated execution mode to a process, the page reference cited by the Examiner *(Coulouris, page 168)* discloses that a process involves the "Creation of an execution environment" with explicitly "defining an initializing the address space of a newly created process". Thus the fact that the process contains is a mode of execution containing its own address space make it "isolated" in this respect.

A process, as understood in the art of computer science, particularly with respect to operating systems, is not a typical process as understood by ordinary people. More specifically a process concerning the technical arts of computer science embodies a "mode of execution" running on a particular processor and operating system. A process has allocated to it, its own page tables, address space, values set for variables, and other resources that other processes cannot typically access. It is a separate entity defined by logical boundaries as set by the operating system in which all the resources necessary for executing a particular process are allocated.

This isolation is essential to the functioning value of the process. The fact that processes contain their own memory address space and own resources is what allows the operating system running two or more concurrent processes to be reasonably certain that they can both execute without interfering in each other's operations.

For Example, suppose the Examiner were running two programs on his computer. Internet Explorer and Netscape. Both of these programs must be stored in memory somewhere. Suppose the Examiner then chooses to alter his font size settings in Internet Explorer which are stored in arbitrarily suggested memory address 0055. Suppose however that Netscape also uses memory address 0055 for toolbar settings. The result would be that alteration of memory address 0055 in Internet Explorer for font size settings would also cause a change in the toolbar setting for Netscape. For this reason, the process was devised as a logical mechanism for use in which each program would then have set aside, its own specific memory pages and resources and would be executed with respect to these resources.

Thus, while Coulouris does not explicitly use the term "isolated" with regards to the execution environment, it is evident from the function of the process that isolation of this execution mode is indeed present and essential in value to the typical functionality of processes. This knowledge would also be evident to one of ordinary skill in the art. For this reason, the Examiner has asserted for the rejection of claim 1, that a process is indeed an "isolated mode of execution"

Applicant has additionally argued the following:

Second, although Coulouris discusses software interrupts at p.172, it is in the context of comparing thread context switches with process context switches. Coulouris does not disclose identifying if an event is one of a class of events to be handled in the isolated execution mode, nor handling the event using the first page table map if the event is identified as one of the class of events to be handled in the isolated execution mode.

The Examiner contends that the event to be handled by the process (or a thread within that process) implies that a determination has been made that the event is one of class which is to be handled by that process, where the "class" is the event belonging to that process. Coulouris p. 172 has disclosed that threads within a process are capable of handling interrupts or events.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coulouris et al. and Silberschatz et al.

In reference to claim 1:

(Coulouris et al. Section 6.3 Processes and Threads) discloses a method comprising:

- Identifying if an event is one of a class of events to be handled in the isolated execution mode, where the isolated execution mode is a processor running a secure process (Page 168), and the event is one of an event or events that might be handled by that process, where threads within a process have their own software interrupt handling mechanisms
- Handling the event using the first page table map if the event is identified as one of the class of events to be handled by the isolated execution mode, where the first page table map is the virtual memory map which maps the memory for the running processes(page 169, 190-192), and the event identified as one of the events to be handled by the isolated execution mode is an event that is to be handled by that process. (page 172)

Coulouris et al. does not explicitly disclose

- Maintaining a first page table map for use in an isolated execution mode and a second page table map for use in a normal execution mode.
- Dynamically swapping between the first page table map and the second page table map responsive to a change in execution mode.

Silberschatz et al. (p 270-271) discloses

- Maintaining a first page table map for use in an isolated execution mode and a second page table map for use in a normal execution mode, where the first page table map is a standard process which executes its own code in an isolated manner, and the normal execution mode is the special case of shared pages between processes.
- Dynamically swapping between the first page table map and the second page table map responsive to a change in execution mode, where processes are isolated execution modes and changing from one execution mode to another would involve a context switch from one process that doesn't use shared pages to another that does. P. 92 (processes)

Silberschatz et al. (p 270-271) discloses that there is an advantage to sharing common code, particularly in the context of a time-sharing environment, and that reentrant shared code can result in a significant savings of total memory space. P. 271 (paragraph 2)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the shared code processes of Silberchatz et al. with the isolated execution processes of Coulouris et

al. in order to allow for significant savings in memory while still retaining the logical boundaries of the process to allow for managed concurrent execution.

In reference to claim 3:

Coulouris et al. and Silberschatz et al. discloses the method of claim 1 wherein dynamically swapping comprises:

- Loading a set of control registers selected based on an exception vector of the event, where a set control registers may be found with the data loaded from the interrupt descriptor table registers in the case of an event, where the control registers are the memory addresses of specialized interrupt handlers which are controlled by the event (exception) table. Silberschatz et al. page (402-404)

In reference to claim 4:

Coulouris et al. and Silberschatz et al. fail to explicitly disclose the method of claim 3 wherein the set of control registers comprises:

- A global descriptor table register
- An interrupt descriptor table register
- A page table map base address register.

The examiner takes official notice that a global descriptor table register and an interrupt descriptor table register were well known in the art at the time of the invention. In particular a

GDTR and an IDTR are registers that contain entries which associate each interrupt or exception identifier with a descriptor for the set of instructions that are to service the event.
Both of these registers are disclosed in a number of processors and processor programming manuals include the well known 80386 Programmer Reference Manual.

It would have been obvious to one of ordinary skill in the art at the time of invention to have a GDT register and an IDT register, so that processor knows which set of instructions to use to respond to a particular event.

In reference to claim 5:
Coulouris et al. and Silberschatz et al. discloses the method of claim 1 wherein maintaining comprises:

- Mirroring a page table base address register.
- Mirroring a memory map is not explicitly disclosed however,

Silberschatz et al.(page 445) discloses a RAID organization called mirroring in which the whole disk is duplicated. While costly, the advantages of this allow reading that is twice as fast. Silberschatz et al(p. 289) also discloses that memory maps, page tables, and processes may be placed on the actual hard disk itself in virtual memory. Silberschatz et al. discloses on p. 293, Figure 9.3 that page tables and memory maps for the memory may be stored in the actual hard disk.

The mirroring a hard disk containing virtual memory on it as disclosed by Silberschatz et al. inherently discloses

- Mirroring a page table base address register.
- Mirroring a memory map is not explicitly disclosed however,

In reference to claim 6:

(Coulouris et al. Section 6.4 Naming and Protection) discloses the method of claim 1 further comprising:

Defining a set of events that should be handled in isolated execution mode, where the set of events that should be handled by the isolated execution mode are the set of events that should be handled by a particular running process, selected by the server.

In reference to claim 7:

(Coulouris et al. Section 10.4 Distributed Coordinarion) discloses the method of claim 6 wherein the set of events to be handled in the isolated execution mode comprises:

machine check events and clock events, where the machine and clock events involve the synchronization of system clocks in a distributed system.

In reference to claim 8:

Coulouris et al. discloses the method of claim 2 wherein handling comprises:

- Determining if a current mode is the isolated execution mode, where the current mode is determined if it is in isolated execution mode, if it is determined that an isolated process is currently running. (Section 6.4 Naming and Protection)
- Loading a set of control registers with values corresponding to the first page table map if the current mode is not the isolated execution mode and the event is one of the class, where the set of control registers are loaded which contain the descriptor for the set of instructions needed to handle the current event, if it is found that the event is not to be handled by the current running process, but by another process. (Section 6.4 Naming and Protection)
- Dispatching an exception vector after the loading is complete, where the exception vector for the event is be dispatched once the new process capable of handling the event is loaded or switched to. (Section 6.4 Naming and Protection) & Figure 6.12

Claim 9 is rejected for the same reasons as claim 5, where a selection unit to select which page table map is applied responsive to receipt of an event is disclosed by (Section 6.4, Figure 6.12)

Claim 10 is rejected for the same reasons as claim 9.

In reference to claim 12:

Coulouris et al. and Silberschatz et al. disclose a platform comprising:

- A processor executing in one of normal execution mode and isolated execution mode, where the processor is inherently present and necessary to execute the instructions of a

process (Coulouris et al. p. 168), and where the isolated execution mode is a standard process, while a normal execution mode comprising shared memory in nonisolation is disclosed using shared pages. Silberschatz et al. (p. 270 – 272)

- A first set of control registers to define a current memory map of the platform, where the CPU contains registers containing the process IDs and logical addresses of the process control blocks. Silberschatz et al. (p. 264, figure 8.16 and page 270, figure 8.20)
- A mapping unit to dynamically load the first set of control registers responsive to an event if the event should be handled using an alternate memory map, where the alternate memory map the sharing of reentrant code between two processes. Silberschatz et al. (p. 270 – 272), while a context switch to another process is responsive to an interrupt. Silberschatz et al. (p. 92, Figure 4.3)

In reference to claim 13:

Coulouris et al. discloses the platform of claim 12 wherein the mapping unit comprises:

- A second set of registers having a first subset corresponding to control register values for a normal execution mode memory map and a second subset corresponding to control register values for an isolated execution mode memory map, where an isolated execution mode memory map is the memory map that is contained by the virtual memory map, the kernel map for the processes, and where the normal execution mode has set of registers for a shared memory map. (Section 6.5 and Memory sharing)

- A selection unit to select between the first subset and the second subset, where the selection unit selects an alternate isolated process to perform execution if it is found necessary to handle the clients' request. (Section 6.4 and Figure 6.12)

Claim 14 is rejected for the same reasons as claim 3.

Claim 15 is rejected for the same reasons as claim 4.

Conclusion

6. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of the final action and the advisory action is not mailed under after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension pursuant to 37 CFR 1.136(A) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

7. Any inquiry concerning this communication from the examiner should be directed to Thomas M Ho whose telephone number is (571)272-3835. The examiner can normally be reached on M-F from 9:30 AM - 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gregory A. Morse can be reached on (571)272-3838.

The Examiner may also be reached through email through Thomas.Ho6@uspto.gov

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571)272-2100.

General Information/Receptionist	Telephone: 571-272-2100	Fax: 703-872-9306
Customer Service Representative	Telephone: 571-272-2100	Fax: 703-872-9306

TMH

April 1st, 2005

GILBERTO BARRON JR.
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100